ELSEVIER

Available at WWW.MATHEMATICSWEB.ORG POWERED BY SCIENCE DIRECT

Journal of Computational and Applied Mathematics 152 (2003) 481–491

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Aspects for a block version of the interval Cholesky algorithm

Uwe Schäfer

Institut für Angewandte Mathematik, Universität Karlsruhe, D-76128 Karlsruhe, Germany

Received 20 November 2001; received in revised form 3 June 2002

Abstract

Using the domain decomposition method we give an application of a block version of the interval Cholesky algorithm. We give an example where this version accelerates the calculation of bounds for the solutions of linear systems of equations with symmetric matrices and right-hand sides both of which are varying within given intervals.

Keywords: Interval arithmetic; Cholesky algorithm; Domain decomposition method; H-matrices

1. Introduction

Let $\mathbb{R}$ denote the reals and let $\mathbb{IR}$ denote the set of all compact intervals $[a] := [\underline{a}, \overline{a}] = \{a \in \mathbb{R} : \underline{a} \leqslant a \leqslant \overline{a}\}$. We consider an interval matrix $[A]$ with n columns and n rows (we write $[A] \in \mathbb{IR}^{n\times n}$), i.e., we consider a matrix with intervals as its elements, and we consider an n-dimensional interval vector $[b]$ (we write $[b] \in \mathbb{IR}^n$). Then an interval matrix $[A] \in \mathbb{IR}^{n\times n}$ can be expressed as

$$[A] = ([a_{ij}]) = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{n\times n} : \underline{A} \leqslant A \leqslant \overline{A}\},$$

where the $\leqslant$-sign is meant componentwise.

We are interested in the so-called symmetric solution set

$$S_{\mathrm{sym}} := \{x \in \mathbb{R}^n : Ax = b,\ A = A^{\mathrm{T}},\ A \in [A],\ b \in [b]\}.$$

Since it is difficult to get S_{sym} (see [2]) one is satisfied to find an interval vector $[x]$ satisfying $S_{\mathrm{sym}} \subseteq [x]$, for the case that S_{sym} is bounded. A well-known algorithm to get such an $[x]$ is the interval Cholesky algorithm (see [1]).

E-mail address: uwe.schaefer@math.uni-karlsruhe.de (Uwe Schäfer).

0377-0427/03/$ - see front matter
PII: S0377-0427(02)00725-2

However, in some applications (see Section 4) $[A]$ and $[b]$ are given with a fixed partition, i.e.,

$$[A] = \begin{pmatrix} [A_{11}] & [A_{12}] & \dots [A_{1k}] \\ [A_{21}] & [A_{22}] & \dots [A_{2k}] \\ \vdots & \vdots & \vdots \\ [A_{k1}] & [A_{k2}] & \dots [A_{kk}] \end{pmatrix}, \qquad [b] = \begin{pmatrix} [b_1] \\ \vdots \\ [b_k], \end{pmatrix}, \tag{1}$$

where $[A_{ij}] \in \mathbb{IR}^{n_i \times n_j}$, $[b_i] \in \mathbb{IR}^{n_i}$ and $n_1 + \cdots + n_k = n$. In particular all diagonal blocks are quadratic. So, one is interested in a block version of the interval Cholesky algorithm. In [7], two block interval Cholesky algorithms are introduced. They were called the practical block interval Cholesky algorithm and the theoretical block interval Cholesky algorithm. In this paper we will compare the practical block interval Cholesky algorithm with the interval Cholesky algorithm and we will give an application where the practical block interval Cholesky algorithm should be preferred.

2. The algorithms

Let $[A] \in \mathbb{IR}^{n \times n}$ satisfying $[A] = [A]^{\mathrm{T}}$, i.e., $\underline{A} = \underline{A}^{\mathrm{T}}$ and $\overline{A} = \overline{A}^{\mathrm{T}}$. Furthermore, let $[b] \in \mathbb{IR}^n$ be given. In [1], the following algorithm was introduced:

{The interval Cholesky algorithm}
{ Step 1. $[L][L]^{\mathrm{T}}$ decomposition}
for $j := 1$ **to** n **do**

$$[l_{jj}] := \sqrt{[a_{jj}] - \sum_{i=1}^{j-1} [l_{ji}]^2} \quad \{ \text{ where } [l_{ji}]^2 = \{ l_{ji}^2 : l_{ji} \in [l_{ji}] \} \};$$

for $v := j + 1$ **to** n **do**

$$[l_{vj}] := \left([a_{vj}] - \sum_{i=1}^{j-1} [l_{ji}][l_{vi}] \right) \Big/ [l_{jj}];$$

{ Step 2. Interval forward substitution }
for $j := 1$ **to** n **do**

$$[y_j] := \left([b_j] - \sum_{i=1}^{j-1} [l_{ji}][y_i] \right) \Big/ [l_{jj}];$$

{ Step 3. Interval backward substitution }
for $j := n$ **downto** 1 **do**

$$[x_j] := \left([y_j] - \sum_{i=j+1}^{n} [l_{ij}][x_i] \right) \Big/ [l_{jj}];$$

ICH$([A], [b]) := [x]$.

We will need the following notation:

- If the interval Cholesky algorithm is feasible for $[A] \in \mathbb{IR}^{n\times n}$ and $[b] \in \mathbb{IR}^n$, then we call $([L], [L]^{\mathrm{T}})$ the interval Cholesky decomposition of $[A]$. We define $\mathrm{IChol}([A]) := [L]$.
- Step 2 will be abbreviated by $[y] := \mathrm{IFS}([L], [b])$.
- Step 3 will be abbreviated by $[x] := \mathrm{IBS}([L]^{\mathrm{T}}, [y])$.
- If $[L] \in \mathbb{IR}^{n\times n}$ is a lower triangular matrix satisfying $0 \notin [l_{ii}],\ i = 1, \ldots, n$, and if $[B] \in \mathbb{IR}^{n\times v}$, then we define

$$\mathrm{IFSM}([L], [B]) := (\mathrm{IFS}([L], [B]_{.1}), \ldots, \mathrm{IFS}([L], [B]_{.v})) \in \mathbb{IR}^{n\times v}.$$

By $[B]_{.i}$ we denote the ith column of $[B]$.

Let $[A] \in \mathbb{IR}^{n\times n}$ satisfying $[A] = [A]^{\mathrm{T}}$ and $[b] \in \mathbb{IR}^n$ be given with a fixed partition (1). In [7], the following algorithm was introduced:

{The practical block interval Cholesky algorithm}
{Step 1. $([L], [L]^{\mathrm{T}})$ decomposition }
for $j := 1$ **to** k **do**

$$[L_{jj}] := \mathrm{IChol}\left([A_{jj}] - \sum_{i=1}^{j-1} [L_{ji}][L_{ji}]^{\mathrm{T}}\right);$$

for $l := j + 1$ **to** k **do**

$$[L_{lj}] := \left(\mathrm{IFSM}([L_{jj}], [A_{jl}] - \sum_{i=1}^{j-1} [L_{ji}][L_{li}]^{\mathrm{T}})\right)^{\mathrm{T}};$$

{ Step 2. Block interval forward substitution }
for $j := 1$ **to** k **do**

$$[y_j] := \mathrm{IFS}([L_{jj}], [b_j] - \sum_{i=1}^{j-1} [L_{ji}][y_i]);$$

{ Step 3. Block interval backward substitution }
for $j := k$ **downto** 1 **do**

$$[x_j] := \mathrm{IBS}([L_{jj}]^{\mathrm{T}}, [y_j] - \sum_{i=j+1}^{k} [L_{ij}]^{\mathrm{T}}[x_i]);$$

$\mathrm{PractBlIchol}([A], [b]) := [x]$.

Remark 2.1. If $[l_{ji}]^2$ is substituted by $[l_{ji}] \cdot [l_{ji}]$ within the interval Cholesky algorithm, then the latter is identical to the practical block interval Cholesky algorithm concerning the partition $k = n,\ n_1 = \cdots = n_k = 1$.

3. Feasibility results

Let $[A] \in \mathbb{IR}^{n\times n}$ and $[b] \in \mathbb{IR}^n$. Suppose the interval Cholesky algorithm and the practical block interval Cholesky algorithm are feasible, then we have

$$S_{\mathrm{sym}} \subseteq \mathrm{ICH}([A],[b]) \quad \text{and} \quad S_{\mathrm{sym}} \subseteq \mathrm{PractBlIchol}([A],[b]).$$

See [1,7], respectively. In addition, it is easy to see that

$$S_{\mathrm{sym}} \subseteq \mathrm{ICH}([A],[b]) \subseteq \mathrm{PractBlIchol}([A],[b]) \tag{2}$$

holds taking into account $[l_{ji}]^2 \subseteq [l_{ji}] \cdot [l_{ji}]$.

It can even happen that the interval Cholesky algorithm is feasible, but the practical block interval Cholesky algorithm is infeasible.

Example 3.1. Let us consider

$$[A] = \begin{pmatrix} 1 & [-1,1] & 0 & 0 \\ [-1,1] & 2 & [\sqrt{2},2] & [\sqrt{2},2] \\ 0 & [\sqrt{2},2] & 5 & 6 \\ 0 & [\sqrt{2},2] & 6 & 4+\left(\frac{16}{3}\right)^2 \end{pmatrix}.$$

The interval Cholesky algorithm leads to

$$\begin{pmatrix} [l_{11}] & 0 & 0 & 0 \\ [l_{21}] & [l_{22}] & 0 & 0 \\ [l_{31}] & [l_{32}] & [l_{33}] & 0 \\ [l_{41}] & [l_{42}] & [l_{43}] & [l_{44}] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ [-1,1] & [1,\sqrt{2}] & 0 & 0 \\ 0 & [1,2] & [1,2] & 0 \\ 0 & [1,2] & [1,5] & \left[\frac{\sqrt{31}}{3}, \sqrt{2+\left(\frac{16}{3}\right)^2}\right] \end{pmatrix},$$

whereas the practical block interval Cholesky algorithm concerning the partition $k=4$, $n_1=\cdots n_4=1$ results in

$$[L] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ [-1,1] & [1,\sqrt{3}] & 0 & 0 \\ 0 & \left[\sqrt{\frac{2}{3}},2\right] & \left[1,\sqrt{\frac{13}{3}}\right] & 0 \\ 0 & \left[\sqrt{\frac{2}{3}},2\right] & \left[2\sqrt{\frac{3}{13}},\frac{16}{3}\right] & \left[0,\sqrt{\left(\frac{16}{3}\right)^2+\frac{10}{3}-\frac{12}{13}}\right] \end{pmatrix}.$$

So, for arbitrary $[b] \in \mathbb{IR}^n$ the practical block interval Cholesky algorithm will break down in Step 2 due to division by an interval containing 0.[1]

The situation described in Example 3.1 cannot arise if H-matrices are considered (see below).

Definition 3.1. Concerning an interval matrix $[A] \in \mathbb{IR}^{n\times n}$ we associate the comparison matrix $\langle [A] \rangle = (c_{ij}) \in \mathbb{R}^{n\times n}$ defined by

$$c_{ij} := \begin{cases} -\max\{|a_{ij}| : a_{ij} \in [a_{ij}]\} & \text{if } i \neq j, \\ \min\{|a_{ij}| : a_{ij} \in [a_{ij}]\} & \text{if } i = j. \end{cases}$$

If $\langle [A] \rangle^{-1}$ exists and if $\langle [A] \rangle^{-1} \geqslant 0$, then $[A]$ is called an H-matrix (see [5]).

Theorem 3.1. *Let* $[A]^{\mathrm{T}} = [A] \in \mathbb{IR}^{n\times n}$ *be an* H*-matrix satisfying* $0 < \underline{a}_{ii}$, $i = 1, \ldots, n$. *Furthermore, let* $[b] \in \mathbb{IR}^n$ *be given. Then the following holds*:

1. *The interval Cholesky algorithm is feasible.*
2. *The practical block interval Cholesky algorithm is feasible for any partition of* $[A] \in \mathbb{IR}^{n\times n}$; *in particular, the interval Cholesky algorithm is feasible even if* $[l_{ji}]^2$ *is substituted by* $[l_{ji}] \cdot [l_{ji}]$.

For the proofs we refer to [1,7], respectively.

4. Domain decomposition method

Due to (2) one may think that there is no need for the practical block interval Cholesky algorithm. But this is not true as we will see in this section.

Example 4.1. Let us consider

$$[A] = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & [4,9] & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}, \quad [b] = \begin{pmatrix} 2 \\ -3 \\ [-1,1] \\ [2,4] \\ 0 \end{pmatrix}.$$

[1] One can show that it is not possible to find an interval matrix $[A] \in \mathbb{IR}^{n\times n}$ with $n \in \{1,2,3\}$ such that the interval Cholesky algorithm is feasible, but the practical block interval Cholesky algorithm is infeasible.

It holds

$$\mathrm{ICH}([A],[b]) = \begin{pmatrix} \left[\frac{94+13\sqrt{10}}{940}, \frac{1444-17\sqrt{10}}{2550}\right] \\ \left[\frac{13\sqrt{10}-846}{470}, \frac{-1106-17\sqrt{10}}{1275}\right] \\ \left[\frac{39\sqrt{10}-658}{940}, \frac{594-17\sqrt{10}}{850}\right] \\ \left[\frac{65}{94}, \frac{198}{85}\right] \\ \left[\frac{65}{94}, \frac{198}{85}\right] \end{pmatrix} \approx \begin{pmatrix} [0.1437336, 0.5451926] \\ [-1.7125327, -0.9096146] \\ [-0.5687991, 0.6355779] \\ [0.6914893, 2.3294118] \\ [0.6914893, 2.3294118] \end{pmatrix}.$$

If the third column of $[A]$ is exchanged with the fifth column and if the third row of $[A]$ and of $[b]$ is exchanged with the fifth row, then we consider

$$[\tilde{A}] = \begin{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} & 0 & \begin{pmatrix} 0 \\ -1 \end{pmatrix} \\ 0 & \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} \\ \begin{pmatrix} 0 & -1 \end{pmatrix} & \begin{pmatrix} 0 & -1 \end{pmatrix} & [4,9] \end{pmatrix}, \quad [\tilde{b}] = \begin{pmatrix} 2 \\ -3 \\ 0 \\ [2,4] \\ [-1,1] \end{pmatrix}.$$

Concerning the partition $k=3,\ n_1=n_2=2,\ n_3=1$ we have

$$\mathrm{PractBlIchol}([\tilde{A}],[\tilde{b}]) = \begin{pmatrix} \left[\frac{9}{51}, \frac{27}{51}\right] \\ \left[-\frac{84}{51}, -\frac{48}{51}\right] \\ \left[\frac{13}{17}, \frac{39}{17}\right] \\ \left[\frac{13}{17}, \frac{39}{17}\right] \\ \left[-\frac{8}{17}, \frac{10}{17}\right] \end{pmatrix} \approx \begin{pmatrix} [0.1764705, 0.5294117] \\ [-1.647058, -0.941176] \\ [0.7647058, 2.2941176] \\ [0.7647058, 2.2941176] \\ [-0.4705882, 0.5882352] \end{pmatrix}.$$

Rearranging the unknowns (i.e., the third component of $\mathrm{PractBlIchol}([\tilde{A}],[\tilde{b}])$ is exchanged with the fifth component) we get an inclusion of $S_{\mathrm{sym}} := \{x \in \mathbb{R}^n : Ax = b, A = A^{\mathrm{T}},\ A \in [A],\ b \in [b]\}$ which is closer than $\mathrm{ICH}([A],[b])$.

The technique used in Example 4.1 is called domain decomposition method [6]. It is used to transform a tridiagonal (or in general a band) matrix A into a block arrowhead matrix

$$\tilde{A} = \begin{pmatrix} A_1 & 0 & \dots & 0 & B_1 \\ 0 & A_2 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & A_p & B_p \\ B_1^{\mathrm{T}} & \dots & \dots & B_p^{\mathrm{T}} & A_s \end{pmatrix}. \tag{3}$$

This special structure can be exploited by doing the first p block steps in parallel. But even if no parallel computer is used the practical block interval Cholesky algorithm can get results much faster than the interval Cholesky algorithm as we will see in our last example.

4.1. A detailed example

We consider the boundary value problem

$$\begin{aligned} &-y''(x) = f(x, y(x), y'(x)), \quad x \in [a, c], \\ &y(a) = \alpha, \quad y(c) = \beta. \end{aligned} \tag{4}$$

We assume the problem expressed by (4) has a unique solution and that its first four derivatives are continuous.

We also assume f is a rational function of x, y and y'. Then we choose $n \in N$ and subdivide the interval $[a, c]$ into sub-intervals via $h := (c - a)/(n + 1)$, $x_0 := a$ and $x_{i+1} := x_i + h$, $i = 0, 1, \dots, n$.

At each interior meshpoint $x_1, \dots, x_n$ we replace discrete approximations for the derivatives in the differential equation. The error in these approximations can be analytically expressed and then bounded using interval arithmetic (see [3]). Finally, we get an interval vector $[b] \in \mathbb{IR}^n$ and the existence of $b \in [b]$ such that

$$y := \begin{pmatrix} y(x_1) \\ \vdots \\ y(x_n) \end{pmatrix}$$

is the unique solution of $Ax = b$, where $y(x)$ is the unique solution of (4) and

$$A = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n \times n}. \tag{5}$$

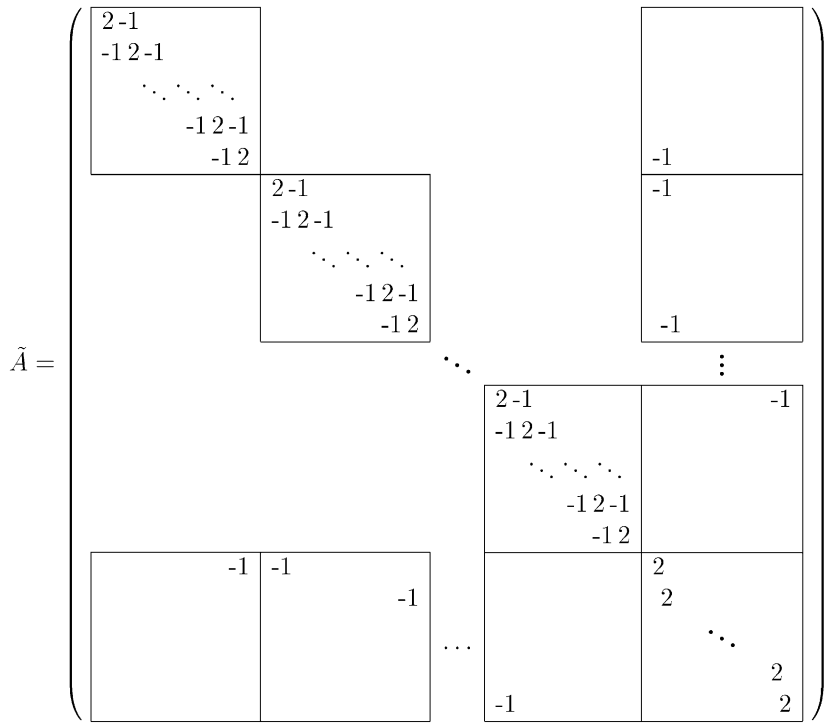

Fig. 1. Discretization matrix after domain decomposition.

On the one hand, since A is an H-matrix (see [5, p. 105]), we can apply Theorem 3.1 to A and $[b]$ and get

$$y \in S_{\text{sym}} \subseteq \text{ICH}(A,[b]).$$

On the other hand, we can apply the domain decomposition method; i.e., we partition the grid points x_i and call D_1 the points $x_1, x_2, \ldots, x_q$, D_2 the points $x_{q+2}, x_{q+3}, \ldots, x_{2q+1}$, and so on. For simplicity, we assume that $n = pq + (p-1)$. Then there are p sets $D_1, \ldots, D_p$ also called domains each containing q points and $p-1$ points $x_{q+1}, x_{2q+2}, \ldots, x_{(p-1)q+(p-1)}$ which are between the domains D_i. These $p-1$ points constitute the separator set S.

If we order the unknowns y_i by numbering those points in the domains first and those points in the separator set last we get as the block arrowhead matrix (3) the matrix given in Fig. 1 (see [6]).

It is easy to verify that $\tilde{A}$ is an H-matrix, too. So, we can apply Theorem 3.1 to $\tilde{A}$ and $[\tilde{b}] = \pi([b])$ and get[2]

$$y \in \pi^{-1}(\text{PractBlIchol}(\tilde{A},[\tilde{b}])).$$

[2] $[\tilde{b}]$ has arisen from $[b]$ by permuting the components according to $\tilde{A}$ and A. (see Example 4.1.) This permutation we want to denote by π.

We have implemented both algorithms using Pascal-XSC [4], where $[b]$ and $[\tilde{b}]$, respectively are defined for simplicity by considering the simple example

$$f(x, y, y') = e^x, \quad \alpha = \beta = 0, \quad a = 0, \quad c = 1,$$

and using the ideas of [3].

Since A and $\tilde{A}$ are sparse we can accelerate the algorithms by neglecting all calculations like $[a] \cdot 0$ within the implementation. The interval Cholesky algorithm reduces to an algorithm, which is denoted by Icholtri(n). It is applied to A of (5) and $[b]$ and its running time is linear with respect to n. (The only input of the program is the dimension n.)

The practical block interval Cholesky algorithm is modified in the following way. The input of the program is p, which is the number of the domains. Then we define $q := p - 1$ [3] and get $n = (p + 1)(p - 1)$. It is $A_1 = \cdots = A_p$. We calculate

$$[L_{11}] = \mathrm{IChol}(A_1) = \begin{pmatrix} [l_{11}] & 0 & \dots & \dots & 0 \\ [l_{21}] & [l_{22}] & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & [l_{q-1q-1}] & 0 \\ 0 & \dots & 0 & [l_{qq-1}] & [l_{qq}] \end{pmatrix}$$

as in Icholtri(n) with n substituted by $n_1 = q = p - 1 \ll n$ and get

$$[L] = \begin{pmatrix} [L_{11}] & 0 & \dots & 0 \\ 0 & [L_{11}] & \ddots & \vdots \\ \vdots & & & \vdots \\ 0 & 0 & [L_{11}] & 0 \\ (\mathrm{IFSM}([L_{11}], B_1))^{\mathrm{T}} & \dots & (\mathrm{IFSM}([L_{11}], B_p))^{\mathrm{T}} & \mathrm{IChol}(\boldsymbol{\Theta}) \end{pmatrix}$$

with

$$\boldsymbol{\Theta} = A_s - \sum_{i=1}^{p} (\mathrm{IFSM}([L_{11}], B_i))^{\mathrm{T}} \cdot \mathrm{IFSM}([L_{11}], B_i).$$

The calculation of $\boldsymbol{\Theta}$ can be accelerated. If $i > 1$, then $-e_1 \in \mathbb{R}^q$ occurs as a column in every B_i, where e_1 denotes the first unit vector. (Note that we are working with B_i and not with B_i^{T}.) Therefore,

[3] In that way the matrices $A_1, A_2, \dots, A_p$ and A_s have the same dimensions and the programming using Pascal-XSC has become easier.

Table 1
Time comparison

	Icholtri (n)	ArrPractIchol (p)	blockinit (p)
$p = 99$ $n = 9800$	27 s	18 s	16 s
$p = 159$ $n = 25,280$	1.10 min	45 s	40 s
$p = 199$ $n = 39,600$	1.44 min	1.11 min	1.02 min
$p = 249$ $n = 62,000$	2.42 min	1.49 min	1.38 min
$p = 299$ $n = 89,400$	3.49 min	2.34 min	2.20 min
$p = 349$ $n = 121,800$	5.27 min	3.37 min	3.07 min

we calculate $[z] := \mathrm{IFS}([L_{11}], -e_1) \in \mathbb{IR}^q$ only once. If $1 < i < p$ we have

$$\mathrm{IFSM}([L_{11}], B_i) = \left(0, \ldots, 0, \underbrace{[z]}_{i-1}, \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1/[l_{qq}] \end{pmatrix}}_{i}, 0, \ldots, 0 \right) \in \mathbb{IR}^{q \times p-1}.$$

$\boldsymbol{\Theta}$ can be calculated very easily. It is a symmetric tridiagonal matrix. The diagonal entries are all of the same value

$$2 - [z]^{\mathrm{T}}[z] - \frac{1}{[l_{qq}] \cdot [l_{qq}]} \tag{6}$$

and the subdiagonal entries are all of the same value $[z_q]/[l_{qq}]$. By substituting $[l_{qq}] \cdot [l_{qq}]$ by $[l_{qq}]^2$ we finally get an algorithm denoted by ArrPractIchol(p). We do not want to go into further details.

We have compared these two algorithms to each other concerning the running time. To strengthen the difference we have implemented a third program which only calculates and prints out $[\tilde{b}]$. This program is denoted by blockinit(p). Some running times are illustrated in Table 1.

One can verify that $\underline{b_i} \geqslant 0$ for all $i = 1, \ldots, n$ holds in our example. Using Theorem 4.11 in [1] in theory we must have Icholtri(n) = ArrPractIchol(p), if $n = (p+1)(p-1)$. But due to rounding

errors there are tiny differences. E.g., ArrPractIchol(3)=

```
[ 7.337647070500889E-002, 7.346376846594350E-002]
[ 1.329437083812990E-001, 1.330964794629347E-001]
[ 1.770803574701071E-001, 1.772767774322102E-001]
[ 2.039745222173048E-001, 2.041927666196417E-001]
[ 2.116013745670812E-001, 2.118196189694181E-001]
[ 1.976981305561438E-001, 1.978945505182469E-001]
[ 1.597360859570702E-001, 1.598888570387060E-001]
[ 9.488936557650776E-002, 9.497666333744240E-002]
```

and Icholtri(8)=

```
[ 7.337647070500878E-002, 7.346376846594344E-002]
[ 1.329437083812987E-001, 1.330964794629346E-001]
[ 1.770803574701068E-001, 1.772767774322100E-001]
[ 2.039745222173046E-001, 2.041927666196414E-001]
[ 2.116013745670809E-001, 2.118196189694178E-001]
[ 1.976981305561436E-001, 1.978945505182467E-001]
[ 1.597360859570701E-001, 1.598888570387058E-001]
[ 9.488936557650770E-002, 9.497666333744229E-002].
```

5. Conclusions

We have shown an example where the practical block interval Cholesky algorithm gets an inclusion of the symmetric solution set faster than the interval Cholesky algorithm and we have shown that we should not overemphasize (2), since sometimes the application of a block version of the interval Cholesky algorithm applied to an interval matrix $[A]$ and an interval vector $[b]$ makes only sense after permuting some columns and rows of $[A]$ and $[b]$ in order to see a block structure.

References

[1] G. Alefeld, G. Mayer, The Cholesky method for interval data, Linear Algebra Appl. 194 (1993) 161–182.
[2] G. Alefeld, V. Kreinovich, G. Mayer, On the shape of the symmetric, persymmetric and skew-symmetric solution set, SIAM J. Matrix Anal. 18 (1997) 693–705.
[3] E. Hansen, On solving two-point boundary-value problems using interval arithmetic, in: E. Hansen (Ed.), Topics in Interval Analysis, Clarendon Press, Oxford, 1969, pp. 74–90.
[4] R. Klatte, U. Kulisch, M. Neaga, D. Ratz, C. Ullrich, Pascal-XSC, Language Reference with Examples, Springer, Berlin, 1992.
[5] A. Neumaier, Interval Methods for Systems of Equations, Cambridge University Press, Cambridge, 1990.
[6] J.M. Ortega, Introduction to Parallel and Vector Solution of Linear Systems, Plenum Press, New York, 1988.
[7] U. Schäfer, Two ways to extend the Cholesky decomposition to block matrices with interval entries, Reliable Comput. 8 (2002) 1–20.